

MÓDULO 7:

COOKIES Y SESIONES

Eduard Lara

INDICE

1. Cookies

2. Sesiones

1. COOKIES Y SESIONES

- ❖ Facilitan el almacenamiento de información relativa a los visitantes de una página, durante una visita o entre visitas (Mensaje bienvenida)
- ❖ El uso de variables sólo es válida durante la visita del usuario y no es una técnica segura.
- ❖ Requisitos:
 - ❖ El servidor debe admitir las cookies.
 - ❖ Las instrucciones para el envío de cookies y creación de sesiones deben al principio de la página, antes de cualquier etiqueta HTML.

1. COOKIES

- ❖ Las cookies son pequeños ficheros que se escriben en el ordenador del cliente.
- ❖ Contienen información que las páginas visitadas pueden recuperar y utilizar en futuras visitas del usuario (parámetros elegidos anteriormente)
- ❖ La información de una cookie se guarda en variables, de la forma `variable="valor"`
- ❖ Se puede establecer la fecha de caducidad de la cookie (tiempo que el ordenador guardará dicha información)

1. COOKIES

Netscape

❖ Guarda todas las cookies en un único archivo, cookies.txt, en el subdirectorio del usuario que se crea en la instalación del programa y que está dentro de otro llamado users.

Explorer

❖ Crea un archivo para cada cookie, en el directorio "Archivos temporales de internet"

"id equipo"@ "Web que inserta la cookie"

["numero de visitas"].txt → edu@elmundo[1].txt

1. COOKIES

❖ Para enviar cookies usando PHP:

```
setcookie("nombre", "valor", "fecha");
```

↓ ↓ ↓
Nombre variable Valor Fecha caducidad cookie

❖ Después de fecha de caducidad, la «galleta» se deshabilita. Si no se establece fecha, la cookie permanecerá en el equipo del usuario.

❖ La variable de la cookie, puede ser leída con

```
$_COOKIE['nombre_variable']
```

si ha sido escrito antes y no ha expirado.

1. COOKIES

```
<?php
    $fechahora = "éste es el primer acceso";
    $contador = 1;
    date_default_timezone_set('Europe/Paris');
    // si la variable está asignada quiere decir que existe la cookie
    if (isset($_COOKIE['fechahora'])){
        // fecha hora del último acceso
        $fechahora = $_COOKIE['fechahora'];
        $contador = $_COOKIE['contador'];
    }
    // actualización de la cookie: se debe hacer antes de
    // mandar cualquier cabecera a la salida
    setcookie("fechahora",date('d/m/Y h:i:s')); // actualiza la cookie
    setcookie("contador",$contador + 1); // actualiza la cookie
?>
```

1. COOKIES

```
<HTML>
  <HEAD> <TITLE>Uso de cookies en PHP</TITLE> </HEAD>
  <BODY>
    <H3>Fecha y hora actual:
    <?php
      echo date('d/m/Y h:i:s') ;
    ?>
    </H3><BR><B> Contenido de la superglobal $_COOKIE</B><BR>
    <?php
      echo "Elemento fechahora: " . $_COOKIE['fechahora'] . "<BR>";
      echo "Elemento contador: " . $_COOKIE['contador'] . " <BR><BR>";
      echo "<BR>La última vez que accedió a la página: $fechahora <BR>";
      echo "<BR>Cantidad de accesos a esta página : $contador <BR>";
    ?>
  </BODY>
</HTML>
```


2. SESIONES

- ❖ Las sesiones son vitales cuando quiere crear un website que permita las siguientes características:
 - Almacenar y mostrar información sobre un usuario
 - Determinar que grupos de usuarios pertenece una persona
 - Utilizar permisos en tu website.
- ❖ Las sesiones PHP consiguen mejores resultados que las Cookies.
- ❖ Almacenan la información en el servidor web, en una localización elegida en unos ficheros especiales.
- ❖ Estos ficheros están conectados al navegador a través del server y un especial ID, llamado "Session ID"
- ❖ Son virtualmente invisibles al usuario.

2. SESIONES

- ❖ Las sesiones ofrecen mayores posibilidades que las «galletas», las cuales son más inseguras
- ❖ Permiten identificar al usuario de forma unívoca y su perfil personalizada
- ❖ Las sesiones se almacenan en el servidor. No se depende de si el usuario ha aceptado las cookies en su ordenador o ha decidido borrarlas.
- ❖ La información obtenida durante la sesión se almacena en las llamadas **variables de sesión**

2. SESIONES

❖ Las variables son guardadas en un directorio del servidor. Se puede indicar mediante:

```
session_save_path("./sesiones");
```

❖ Para iniciar una sesión, al inicio de la pagina se debe de poner:

```
session_start();
```

❖ PHP al crear una sesión, le asigna un número identificativo y un nombre, accesibles con las siguientes instrucciones:

```
echo session_id()  
echo session_name()
```

2. SESIONES

- ❖ Para asignar un nombre especif. a una sesión:

```
session_name('nombre_sesion');
```

- ❖ Para comunicar a otras páginas que deben continuar con la sesión ya iniciada, se debe propagar la sesión → Se le debe pasar el nombre y su identificador.

```
<a href="pagina_siguiete.php?<?php echo  
session_name()."=".session_id()?>">enlace </a>
```

El usuario verá en la barra de direcciones una cadena inteligible de caracteres.

2. SESIONES

❖ Las variables de sesión son guardadas en el servidor mientras dure ésta. Para crear las variables de sesión:

```
$_SESSION[ 'variable' ]="valor";
```

❖ El servidor no puede actualizar una página dinámica que es almacenada en la caché.

```
session_cache_limiter('nocache, private');
```

→ Evita que la página se almacene en la caché del navegador y pueda actualizarse su contenido.

2. SESIONES

Fichero session.php

```
<?php
    session_start();
    if (!$_SESSION["count"])
        $_SESSION["count"] = 0;
    if ($_GET["count"] == "yes")
        $_SESSION["count"] = $_SESSION["count"] + 1;
    echo "<h1>".$_SESSION["count"]."</h1>";
?>
<a href="session.php?count=yes">Click aquí para contar</a>
```

2. SESIONES

DESTRUCCIÓN DE UNA SESIÓN

- ❑ Es necesario destruir una sesión cuando el usuario cierre su navegador.
- ❑ Si se tiene una sesión registrada llamada "access_granted" con el objetivo de determinar si un usuario fue loginado en tu site, para dar seguridad a los usuarios, se debería usar la función `session_destroy()`
- ❑ `session_destroy()` elimina completamente la sesión, eliminando los ficheros de sesión y limpiando cualquier traza de esa sesión.
- ❑ Si se está usando el array superglobal `$_SESSION`, se debe limpiar los valores del array primero, antes de ejecutar `session_destroy`.

2. SESIONES

DESTRUCCIÓN DE UNA SESIÓN

Fichero page1.php

```
<?php
    session_start(); // start the session
    header("Cache-control: private");
    $_SESSION = array();
    session_destroy();
    echo "<strong>Se ha destruído esta sesión </strong><br />";
    if($_SESSION['name']){
        echo "La sesión está todavía activa";
    } else {
        echo "Ok, la sesión ya no está activa! <br />";
        echo "<a href=\"page1.php\"><< Vuelta atrás</a>";
    }
?>
```